

WE CLAIM:

1. Apparatus for processing data, said apparatus comprising:
processing logic operable to perform processing operations under control of
5 program instructions and subject to interruption by a plurality of interrupt events; and
a nested interrupt controller operable to control nested execution of a plurality
of active interrupt handling programs that are started and uncompleted and which is
responsive to priorities associated with respective ones of said plurality of interrupt
handling programs to control pre-emption of one or more active interrupt handling
10 programs by a pending interrupt handling program; wherein
a priority of a given active interrupt handling program is alterable whilst said
given active interrupt handling program is started and uncompleted; and
said nested interrupt controller is operable:
 - (i) to permit a pending interrupt handling program to pre-empt a plurality of
15 active interrupt handling programs if a priority associated with said pending interrupt
handling program is higher than a highest priority associated with any of said plurality
of active interrupt programs; and
20 (ii) to prevent said pending interrupt handling program from pre-empting said
plurality of active interrupt handling programs if said priority associated with said
pending interrupt handling program is less than said highest priority associated with
any of said plurality of active interrupt handling programs.
2. Apparatus as claimed in claim 1, wherein said nested interrupt controller is
operable to prevent said pending interrupt handling program from pre-empting said
25 plurality of active interrupt handling programs if said priority associated with said
pending interrupt handling program is equal to a highest priority associated with any
of said plurality of active interrupt handling programs.
3. Apparatus as claimed in claim 1, wherein a priority of an interrupt handling
30 program is set by a programmable priority level associated with said interrupt
handling program.
4. Apparatus as claimed in claim 1, wherein said pending interrupt handling
program corresponds to one of:

a newly detected interrupt event; and
one or more previously detected interrupt event for which a corresponding interrupt handling program has not started execution.

5 5. Apparatus as claimed in claim 1, wherein said nested interrupt controller is responsive to stored values corresponding to respective ones of a plurality of interrupt events and indicative of:

- a corresponding interrupt handling program being pending;
- a corresponding interrupt handling program being active; and
- 10 a priority level associated with said corresponding interrupt handling program.

6. Apparatus as claimed in claim 1, comprising stack data storage operable to store state data corresponding to processing pre-empted by one or more active interrupt handling programs.

15

7. Apparatus as claimed in claim 6, wherein said highest priority level associated with any of said plurality of active interrupt handling programs is increased when free space within said stack data store falls below a predetermined free space level.

20

8. Apparatus as claimed in claim 7, wherein said highest priority level associated with any of said plurality of active interrupt handling programs is increased to a programmable priority level which is less than a maximum possible priority level.

25

9. Apparatus as claimed in claim 8, wherein said predetermined free space level is set to leave free space within said stack memory for use during pre-emption by interrupt handling programs having a priority higher than said programmable priority level.

30

10. Apparatus as claimed in claim 7, comprising a plurality of processors within a multiprocessor system, said stack data storage being a shared hardware stack used for communication between said plurality of processors.

11. Apparatus as claimed in claim 6, wherein said stack data storage is a stack memory.

12. A method of processing data, said method comprising the steps of:
5 performing processing operations under control of program instructions, said processing operations being subject to interruption by a plurality of interrupt events;
and

10 controlling nested execution of a plurality of active interrupt handling programs that are started and uncompleted and in response to priorities associated with respective ones of said plurality of interrupt handling programs to control pre-emption of one or more active interrupt handling programs by a pending interrupt event; wherein

15 a priority of a given active interrupt handling program is alterable whilst said given active interrupt handling program is started and uncompleted; and

20 said step of controlling is operable:

(i) to permit a pending interrupt handling program to pre-empt a plurality of active interrupt handling programs if a priority associated with said pending interrupt handling program is higher than a highest priority associated with any of said plurality of active interrupt programs; and

25 (ii) to prevent said pending interrupt handling program from pre-empting said plurality of active interrupt handling programs if said priority associated with said pending interrupt handling program is less than said highest priority associated with any of said plurality of active interrupt handling programs.

13. A method as claimed in claim 12, wherein said nested interrupt controller is operable to prevent said pending interrupt handling program from pre-empting said plurality of active interrupt handling programs if said priority associated with said pending interrupt handling program is equal to a highest priority associated with any of said plurality of active interrupt handling programs.

14. A method as claimed in claim 12, wherein a priority of an interrupt handling program is set by a programmable priority level associated with said interrupt handling program.

15. A method as claimed in claim 12, wherein said pending interrupt handling program corresponds to one of:

a newly detected interrupt event; and
one or more previously detected interrupt event for which a corresponding interrupt handling program has not started execution.

5 16. A method as claimed in claim 12, wherein said step of controlling is responsive to stored values corresponding to respective ones of a plurality of interrupt events and indicative of:

- a corresponding interrupt handling program being pending;
- a corresponding interrupt handling program being active; and
- 10 a priority level associated with said corresponding interrupt handling program.

17. A method as claimed in claim 12, wherein stack data storage is operable to store state data corresponding to processing pre-empted by one or more active interrupt handling programs.

15

18. A method as claimed in claim 17, wherein said highest priority level associated with any of said plurality of active interrupt handling programs is increased when free space within said stack data store falls below a predetermined free space level.

20

19. A method as claimed in claim 18, wherein said highest priority level associated with any of said plurality of active interrupt handling programs is increased to a programmable priority level which is less than a maximum possible priority level.

25

20. A method as claimed in claim 19, wherein said predetermined free space level is set to leave free space within said stack memory for use during pre-emption by interrupt handling programs having a priority higher than said programmable priority level.